

基于邻接关系的空间趋势检测算法研究^{*}

文俊浩^{1,2} 李立新^{3,2} 吴中福¹ 吴红艳¹

(重庆大学计算机学院¹ 重庆大学软件学院² 重庆400044) (信息工程大学电子技术学院 郑州450004)³

Algorithms for Spatial Trend Detection Based on Spatial Neighborhood Relations

WEN Jun-Hao^{1,2} LI Li-Xin^{3,2} WU Zhong-Fu¹ WU Hong-Yan¹

(College of Computer Science, Chongqing University, Chongqing 400044)¹

(Faculty of Software Engineering, Chongqing University, Chongqing 400044)²

(College of Electronic Technology, Information Engineering University, Zhengzhou 450004)³

Abstract A regular change of one or more non-spatial attributes can be detected when moving away from a given start object. Moreover, spatial objects are often influenced by their neighbors. And the influence typically decreases or increases more or less continuously with increasing or decreasing distance. Due to the attributes of the neighbors are always similar or associated to each other, spatial trend detection based on spatial neighborhood relations is analyzed to extract useful knowledge in this paper.

Keywords Spatial neighborhood relations, Spatial trend detection, Spatial neighborhood relations

1 引言

随着信息科学技术发展,社会经济各部门生产、收集、存储和处理数据的能力有了极大提高,数据资源日益丰富,数据挖掘为数据分析提供了强有力的工具^[1]。空间数据挖掘源于提取空间数据有用信息和知识的决策应用的需要,这些知识可能是隐含的规则、空间模式或位于空间和非空间数据中的规则^[2]。空间数据挖掘的主要特征是分析空间关系。然而,空间对象的主要特性是:在空间里位于彼此附近的对象倾向于具有相似或相关联的属性,即某特定空间的属性总是联系的,且能由其邻接对象的属性来解释。因此相对于关系数据库里的挖掘,空间数据挖掘算法为了提取有用的知识,需要考虑邻接对象,空间数据的发现过程及可能发现的模式也就比关系数据更复杂,算法效率也相对较低。

空间数据挖掘的框架是基于对象之间的空间邻接关系和空间关系引进的邻接图和邻接路径形成的。数据库原语或空间数据挖掘的基本操作能有效地描述大多数空间数据挖掘算

法,其优点是类似于关系数据库标准语言 SQL,使用标准原语可以促进新的数据挖掘算法的开发并增加其可移植性,且能够开发有效地支持数据库原语的技术(如索引结构),从而促进所有基于数据库原语的数据挖掘算法的发展^[3]。由于相邻空间对象的特性总是具有相似或关联的特点,以邻接关系为基础的算法研究就成为空间趋势检测算法研究的一个重要方向。

2 空间邻接关系、空间邻接图以及空间邻接索引

2.1 空间邻接关系

空间数据挖掘的数据库原语以邻接图和邻接路径的概念为基础,有三种基本类型的空间关系:拓扑、距离和方向关系(如图1所示)。

拓扑关系是建立于边、内部以及相关对象的外部的基础上的,其关系包括: A disjoint B; A meets B; A overlaps B; A equals B; A covers B; A covered-by B; A contains B; A insides B。

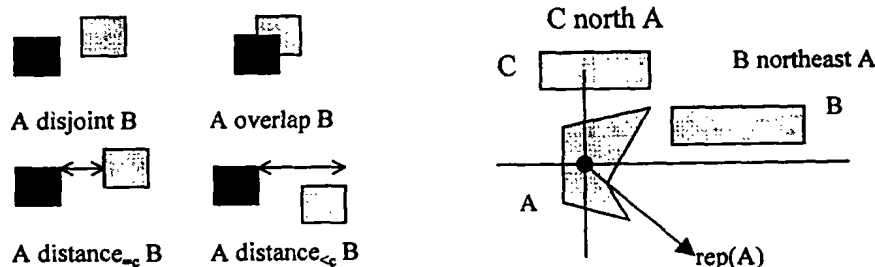


图1 拓扑、距离和方向关系

距离关系用一个数学操作符号和一个确定的常量比较两个对象的距离。如果 dist 是一个距离函数, σ 是一个算术谓词

(如(,)或=), c 是一个实数, 如果 $distance(O_1, O_2) \sigma c$ 成立, 那么在两个空间对象 O_1 和 O_2 之间就有距离关系 $O_1 distance_{\sigma c} O_2$

^{*} 基金项目: 国家十五重大科技计划项目(2002BA107B); 建设部科技攻关项目建科001-4-70; 重庆大学基础及应用基础研究支持项目。文俊浩 讲师, 博士生, 主要研究方向: 数据挖掘、信息网络安全。李立新 副教授, 博士, 主要研究方向: 数据库安全, 数据挖掘。吴中福 教授, 博士导师, 主要研究方向: 计算机网络与通信, 人工智能及其应用。吴红艳 研究生, 主要研究方向: 数据仓库与数据挖掘、决策支持系统。

O_2 成立。

定义方向关系,如 O_2 south O_1 ,选择对象 O_1 的一个有代表性的点作为虚拟坐标系的原点,用它的象限和半平面来定义方向关系。当 O_2 所有的点都定位于平面的各个相关区域才能满足方向谓词。

方向关系不是唯一定义的,但是,对于对象 A 和 B 总有一个最确切的方向关系。例如,若 B 位于 A 东北成立,且 B 位于 A 东方也成立,则 A 和 B 的确切方向关系是 B 位于 A 东北方。对象 A 和 B 的确切的方向关系是唯一确定的。

通过组合基本的空间关系和逻辑操作符,可以定义更复杂的空间关系。例如“ O_1 在 O_2 的北方并且不超过5千米远处”组合了方向关系和距离关系。

2.2 空间邻接图和邻接路径

假设 neighbour 是一个邻接关系,DB 是空间对象的数据库。邻接图 $G_{neighbour}^{DB} = (N, E)$ 是一个 $N = DB$ 个结点和 $E \subseteq N \times N$ 个边的图,其中,如果 neighbour(n_1, n_2)成立,当且仅当边 $e = (n_1, n_2)$ 存在。如果对于所有的点 $n_i \in N, 1 \leq i \leq k$ 有 neighbour(n_i, n_{i+1})成立,则称结点序列 $[n_1, n_2, \dots, n_k]$ 是一条邻接路径,结点数 k 为邻接路径的长度。

假设对于对象集和邻接路径集可以进行关系代数的标准操作,如选择、并、交以及差运算(如,选择 selection (set,

predicate)返回满足谓词 predicate 所有元素集 set)。以下给出用于邻接图和邻接路径的操作:

neighbours: Graphs $\times$ Objects $\times$ Predicates $\rightarrow$ Sets-of-objects
paths: Sets-of-objects $\rightarrow$ Sets-of-paths
extensions: Graphs $\times$ Sets-of-paths $\times$ Integer $\times$ Predicates $\rightarrow$ Sets-of-paths

操作 neighbours (Graphs, Objects, Predicates) 返回在图 graph 里与对象 object 存在路径连接且满足谓词 predicate 的对象集。操作 paths(objects)生成由 objects 里的某元素产生的长度为1的路径。操作 extensions (Graphs, Sets-of-paths, length, Predicates) 返回在 graph 中通过扩展 Sets-of-paths 中的一条路径得到的长度 length 确定且满足谓词 predicate 的所有路径集。如果 paths 不包含元素,则表明一个空的结果集,即 paths 中没有可扩展的元素。

因为邻接路径的数目可能非常大,参数 predicate 在操作 neighbours 和 extensions 里充当过滤器的作用,限制有效的邻接对象和路径的数目。定义 predicate 可以利用对象或路径的空间和非空间属性。空间模式是对象的邻接对象对其产生某种影响的结果,并且,这种影响随着距离的增加和减少而不断地或多或少地增加和减少。为了仅产生相关路径,引进空间筛选谓词,选择所有路径的特定子集,这样可以明显地减少空间数据挖掘算法在邻接路径上的运行时间。

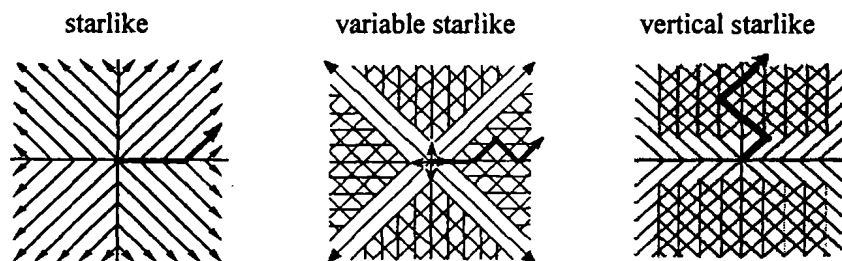


图2 筛选谓词的图例

星形(starlike)筛选仅产生一小部分“粗糙”路径,是一种限制性十分强的筛选方法。这种筛选适合于多种应用,运行效率也较高。可变星形(variable-starlike)的筛选要求当扩展 $edge(n_k, n_{k+1})$ 的时候,必须满足是 p 的初始方向的要求,从而产生更多“细粒度”的路径。可变星形筛选比星形筛选的空间分析更详细,但是因为需要处理更多的路径而增加了数据挖掘算法的运行时间。垂直星形(vertical star like)筛选,在垂直方向比水平方向有较少的限制性。当垂直方向比水平方向需要进行更多的分析的时候,这种方法比较合适(如图2所示)。

2.3 邻接索引

假设 DB 是空间对象集,max、dist 是实数,D 是方向关系,T 是拓扑关系, I_{max}^{DB} 表示最大距离 max 的对于 DB 的邻接索引,定义如下:

$$I_{max}^{DB} = \{(O_1, O_2, dist, D, T) | O_1, O_2 \in DB, O_1 distance =_{dist} O_2 \wedge dist \leq max \wedge O_2 DO_1 \wedge TO_2\}$$

一个邻接索引支持一个邻接图集,如果索引包含这个图的每条边的入口,那么就称一个邻接索引对于给定的邻接图是可采用的。为了找到邻接图的可采用的邻接索引,引入邻接关系 r 的临界距离的概念。一般情况下,邻接关系 r 的临界距离是满足 $O_1 r O_2$ 的一对对象的 O_1 和 O_2 的最大距离。下面的定理用来计算邻接关系的临界距离。

定理1 下面的等式对于一个邻接关系 r 成立:

$$c-distance(r) =$$

$$\begin{cases} 0 & \text{如果 } r \text{ 是一个除了相离外的拓扑关系} \\ d & \text{如果 } r \text{ 是关系 } distance <_d \text{ 或 } distance =_d \\ \infty & \text{如果 } r \text{ 是方向关系,并且关系 } distance >_d \text{ 或相离} \\ \min(c-distance(r_1), c-distance(r_2)) & \text{如果 } r = r_1 \wedge r_2 \\ \max(c-distance(r_1), c-distance(r_2)) & \text{如果 } r = r_1 \vee r_2 \end{cases}$$

如果关系为 r 的邻接图的临界距离不大于最大距离 max,那么可以采用最大距离为 max 的邻接索引。如果两个邻接索引 I_{c1}^{DB} 和 I_{c2}^{DB} 且 $c1 < c2$,那么用 I_{c1}^{DB} 会更有效,因为总的来讲, I_{c1}^{DB} 比 I_{c2}^{DB} 有较少的入口。对于邻接图的最小可采用的邻接索引是指临界距离最小的邻接索引。

利用最小可采用的邻接索引来描述邻接操作算法。index selection,选择一个邻接索引,如果没有可采用的邻接索引,那么就使用一些空间索引结构的标准结构方法操作。Filter step 返回基数集明显小于数据库规模的候选对象集(可能会满足具体的邻接关系)。Refinement step,对于所有的对象,评价邻接关系和补充的谓词 pred,通过评价的对象作为结果邻接关系返回。

为了创建一个邻接索引 I_{max}^{DB} ,在数据库上对邻接关系($O_1 distance =_{dist} O_2 \wedge dist \leq max$)可以通过空间邻接索引有效地进行空间连接。对于空间连接返回的每对对象,需要确定距离、方向关系和拓扑关系。结果以元组的形式($O_1, O_2, Distance, Direction, Topology$)存储在根据属性 O_1 检索的的二叉树的关系库里。

数据库的更新,例如空间对象的插入和删除,需要更新导出的邻接索引。然而邻接索引 $I_{\max}^{DB}$ 仅限于由邻接关系 $A_{distance < \max B}$ 定义的相关对象的邻接,可以通过一个邻接索引(在删除的情况下)或使用一个空间邻接结构(插入时)来十分有效地检索邻接关系。也就是说,当更新一个数据库时,不需要重新再建立一个邻接关系。

邻接索引对于复杂的空间对象(如每个对象有大量的顶点时)以及对象的邻接对象的数目的平均值不是很大时尤其有效。例如,当使用 intersect 作为地理信息系统的谓词时,若空间邻接关系的平均邻接对象数目很大,需要考虑空间邻接索引的规模^[5]。视临界距离 $c\text{-distance}(r)$ 和空间对象分布而定,在最坏的情况下,实现邻接索引的存储需求可能是 $O(n^2)$ 。Ester et al.^[4] 提出了邻接操作的一个价值模型。在系统参数(如访问一页的执行时间)和邻接关系的信息(如邻接对象的平均数目)确定的基础上,这个模型可以预测邻接索引的性能增益和时间增益是否能抵消额外的存储需求。

3 空间趋势检测

空间趋势是指当从一个给定的开始对象 O 开始移动的时候,一个或多个非空间属性的规则变化。现在使用从 O 开

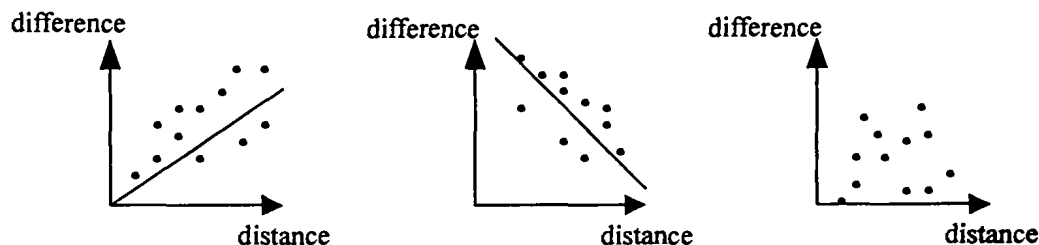


图3 正线性趋势、负线性趋势和没有趋势图示

有时,对 $O_2 \in DB$ 的每个元素得到一对对应值并对整个对应值集进行一次回归分析,在下列的情形下得不到空间趋势:

- 趋势仅出现在局部区域而不是整个数据库。
- 趋势出现在某个方向而不是所有的方向上。

这种情形在实际的数据库中是经常出现的。解决办法是使用邻接路径的概念,并把对象 O_2 定位在从 O_1 开始的某个邻接路径上,当不再发现明显的趋势的时候,就停止扩展这条邻接路径。这样,就限制了对对象的邻接路径的搜索。更重要的是,其中一部分(并不是所有的)邻接路径反映出某种类型的空间趋势就足够了。

算法1发现局部趋势(global trends),即在一个给定的长度区间内的从源 O 开始的所有邻接路径的趋势。算法 detect-global-trends 进行从 O 开始的所有邻接路径的宽度优先的搜索。算法创建同时开始的、相同长度(从 $\min\text{-length}$ 到 $\max\text{-length}$)的所有邻接路径。然后对同样长度的路径集中的每条路径进行回归分析。如果在长度 l 内,没有检测到相关系数 $\text{abs}(\text{correlation}) \geq \min\text{-conf}$ 的趋势,那么就扩展路径长度为 $l+1, l+2, \dots$, 考虑到效率,不再创建 $\max\text{-length}$ 长度的路径。应该选择足够大的 $\min\text{-length}$ 参数,因为只有在一定的最小长度的情况下,趋势才比较明显。算法返回相关系数 $\text{abs}(\text{correlation}) \geq \min\text{-conf}$ 的最大长度的邻接路径集,并且返回了回归函数的斜率和相关系数。斜率描述了当从源对象 O 开始移动时,具体属性变化的程度;斜率较大则表明属性变化较快,而斜率较小则表明在 O 的临域范围内,此属性变化比较

慢。相关系数则用来权衡发现的趋势的置信度:相关系数越大,回归函数的值越能很好地逼近,发现的相应的趋势的置信度也就较高。

假设 g 是一个邻接图, o 是 g 里的一个对象(结点), a 是非空间属性的一个子集, t 是用来回归分析某类型函数,如线形的或指数性的, filter 是邻接路径的筛选函数, $\min\text{-conf}$ 是实数, $\min\text{-length}$ 和 $\max\text{-length}$ 是自然数。空间趋势检测就是发现在 g 里从 o 开始的邻接路径集里的属于某个类型 t 的趋势,其中相关性至少为 $\min\text{-conf}$, 路径满足 filter , 并且长度位于 $\min\text{-length}$ 和 $\max\text{-length}$ 之间。

首先,确定对某个源对象 $O_1 \in DB$ 进行空间趋势检测。为了检测某个非空间属性的规则变化,进行如下的回归分析:自变量 X 表示任一数据库对象 O_2 和源对象之间的距离,因变量 Y 表示 O_1 和 O_2 非空间属性值的变化,那么 X 集和 Y 集包含 DB 的子集 S 的每个元素的一对对应值。如果发现的相关系数的绝对值足够大,则 S 所表明的那一部分 DB 具有从 O 开始的明显的空间趋势。由于各种现象对于邻接对象的影响经常呈线形或者可以转化为线形模型,如指数回归,下面使用线形回归进行分析。图3指出了强正线形趋势(相关系数 $>>0$) 和强负线形趋势(相关系数 $<<0$) 和没有明显趋势的情形。

算法1 局部趋势检测算法

输入:图 $graph; G_r^{DB}$; 对象 $object\ o$; 属性 $attribute\ a$; 回归分析函数类型 $type\ t$;
最小置信度 $real\ \min\text{-conf}$, 最小长度 $integer\ \min\text{-length}$; 最大长度 $\max\text{-length}$; 筛选谓词 $\text{filter}\ f$;
(1) 初始化路径集 $paths$ 为 $\text{extensions}(G_r^{DB}, \{path(o)\}, \min\text{-length}, f)$;
初始化观测值 $observations$ 为空集;
初始化 last_paths , last_correlation 为 NULL;
初始化 first_pos 为 1, last_pos 为 $\min\text{-length}$;
(2) while 路径集不为空 do
for 路径集 $paths$ 里的每条 $path$ do
for $object$ 从 first_pos 到 last_pos do
计算 $\text{diff} = a(object) - a(o)$ 和 $\text{dist} = \text{dist}(object, o)$;
把元组 $\text{tuple}(\text{diff}, \text{dist})$ 插入 $observations$;
对 $observations$ 进行类型为 t 的回归分析;
if 相关系数 $\text{abs}(\text{correlation}) \geq \min\text{-conf}$,
then $\text{last_paths} = paths$;
 $\text{last_correlation} = \text{correlation}$;
 $\text{last_slope} = \text{slope}$; // 回归分析得到的函数的斜率
if $\text{length}(paths) < \max\text{-length}$
then 路径集 $paths$ 为 $\text{extensions}(G_r^{DB}, paths, \text{length}(paths)+1, f)$;
 $\text{last_paths} = \text{last_paths}+1$;
 $\text{first_pos} = \text{last_pos}$;
else 置 $paths$ 为空;
else 返回 $\text{last_paths}, \text{last_correlation}, \text{last_slope}$;
返回 $\text{last_paths}, \text{last_correlation}, \text{last_slope}$;

算法2发现局部趋势,即在一个给定的区间长度内的从源 o 开始的一个邻接路径的趋势。算法 detect-local-trends 以深度优先的方式创建邻接路径。对于每条长度为 $\max\text{-length} \geq$

(下转第138页)

(1) 每一个轮廓线段必须在而且只能在一个基本三角面片中出现。如果上下轮廓线各有 m 和 n 个轮廓线段, 则合理的三角面模型含有 $m+n$ 个基本三角面片。

(2) 如果一个跨距在某一基本三角面中为左跨距, 则该跨距是且仅是另一个三角面片的右跨距。

E. Keppel^[11]和 H. Fuchs^[10]将该问题归结为对有向图的搜索。如图7所示, 可以用一个有 m 行, n 列的有向图 $G(V, A)$ 来表示点列及其连接关系。图中每一个结点表示上轮廓线和下轮廓线之间的一个跨距, V_{ij} 表示点 P_i 到 Q_j 之间的跨距, 有向弧 $[V_{ij}, V_{i,j+1}]$ 称为水平弧, 对应了一个基本三角面片, 其左跨距为 V_{ij} , 右跨距为 $V_{i,j+1}$ 。称 $[V_{ij}, V_{i+1,j}]$ 为垂直弧, $[V_{ij}, V_{i+1,j}]$ 将第 i 行的结点与第 $i+1$ 行的结点连接起来, 它同样对应了一个基本三角面片。 $G(V, A)$ 由跨距节点和水平弧和垂直弧构成。

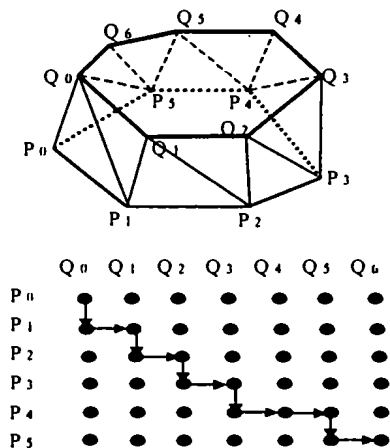


图7 对应于曲面重构问题的有向图搜索问题

因此, 一个可接受的三角面模型对应从 V_{00} 到 $V_{m-1,n-1}$ 的路径。在这样的基础上, 有两种不同的方法来确定所需要的可接受表面的路径: 基于全局搜索策略的优化方法和基于局部计算和决策的启发式方法^[9]。基于全局搜索策略的优化方法有表面积最小, 或体积最大, 或两轮廓之间的跨接边之和最短 (周长最短) 等等, 不论采用哪种具体的算法, 避免不了轮廓拼

接的迭代运算, 效率仍然不高^[9], 因此在本文的三维重建中, 为了减少运算量和数据量, 以提高显示速度, 采用了轮廓的多边形简化和最短对角线法 (一种基于局部计算和决策的启发式方法), 其中, 轮廓多边形简化的基本方法是: 在轮廓检测时, 首先建立轮廓的方向链码; 然后依次搜索方向链码, 其方向改变的点就是多边形的角点, 而方向没有改变的点就是多边形边上的点; 最后保留角点, 去掉边上的点; 于是, 这样就把轮廓简化成了多边形, 并去掉了轮廓直线边上的冗余点, 这样处理, 既减少了轮廓拼接的三角形面片的数量, 又不改变表面的拼接形状, 最短对角线法采用跨接边长度最短, 而不是采用跨接边长度之和最短来作为优化准则。这样就可以在不超过 $(m+n)$ 步的计算中得出两轮廓之间用到的一系列三角面连接的近似最优解, 避免了轮廓拼接的迭代运算, 计算量小, 提高了运算速度, 并且不影响总体的轮廓拼接的形状。

参考文献

- 1 Herman G T, Liu H K. Three-dimensional display of human organs from computed tomograms[J]. Computer Graphics Image Processing, 1979, 9(1): 1~21
- 2 Lorensen W E, Cline H E. Marching cubes: A high resolution 3D surface construction algorithm[J]. Computer Graphics, 1987, 21(4): 163~169
- 3 Cline H E, Lorensen W E. Two algorithms for the 3D reconstruction of tomograms[J]. Medical Physics, 1988, 15(3): 225~233
- 4 Keppel E. Approximating complex surface by triangulation of contour lines[J]. IBM J. Research and Development, 1975, 19(1): 2~11
- 5 Kehtarnavaz N, de Figueiredo R J P. A framework for surface reconstruction from 3D contours [J]. Computers Vision Graphics Image Processing, 1998, 42(1): 32~47
- 6 Chang L W, Chen H W, Ho J R. Reconstruction of 3D medical images: A nonlinear interpolation technique for reconstruction of 3D medical images [J]. Computer Vision Graphics Image Processing: Graphical Models and Image Processing, 1991, 53(4): 298~309
- 7 Meyers D, Skinner S, Sloan K. Surfaces from contours [J]. ACM Transactions on Graphics, 1992, 11(3): 228~258
- 8 管伟光. 体视化技术及其应用[M]. 北京: 电子工业出版社, 1998. 30~32
- 9 唐泽圣等著. 三维数据场可视化[M]. 北京: 清华大学出版社, 1999. 188~190
- 10 Fuchs H, Kedem Z M, Uselton S P. Optional Surface Reconstruction from Planar Contours, Communication of the ACM, 1997, 20(10): 693~702
- 11 Keppel E. Approximating Complex Surfaces by Triangulation of Contour Lines. IBM Journal Res. Develop, Jan. 1995. 2~10

(上接第125页)

$length \geq min-length$ 的邻接路径进行回归分析。当检测到明显的趋势 (即结果回归函数的相关系数 $abs(correlation) \geq min-conf$) 时, 路径进一步扩展。算法返回有明显的空间趋势的两个路径集: 一个为正趋势集, 一个为负趋势集。

算法2 全局趋势检测算法

输入: 图 $graph G^B$; 对象 object o ; 属性 attribute a ; 回归分析函数类型 $type t$;
最小置信度 $real min-conf$; 最小长度 $integer min-length$;
最大长度 $max-length$; 筛选谓词 $filter f$;
(1) 初始化路径集 $paths$ 为 $extensions(G^B, \{path(o)\}, min-length, f)$;
初始化 $positive-trends$ 和 $negative-trends$ 为两个空集;
(2) while 路径集不为空 do;
 初始化观测值 $observations$ 为空集;
 删除路径集 $paths$ 里的下一个元素, 并把它作为 $path$;
 for 从 $min-length$ -th 对象的 $path$ 到最后一个对象的 $path$
 计算 $diff = a(object) - a(o)$ 和 $dist = dist(object, o)$;
 把元组 $tuple(diff, dist)$ 插入 $observations$;
 对 $observations$ 进行类型为 t 的回归分析;
 if 相关系数 $abs(correlation) \geq min-conf$,
 then
 if $correlation > 0$
 then 把元组 $tuple(path, correlation)$ 插入 $positive-trends$;
 else 把元组 $tuple(path, correlation)$ 插入 $negative-trends$;

if $length(paths) < max-length$

then $extensions(G^B, paths, length(paths) + 1, f)$ 加入 $paths$ 的首部;

返回 $positive-trends, negative-trends$;

结论 本文介绍了在邻接关系基础上的空间趋势检测的算法描述, 该算法用了数据库原语和空间数据挖掘的基本操作, 然后针对相邻空间对象的特性总是相似或相关联的特点, 以邻接关系为基础对空间趋势检测进行了分析。

参考文献

- 1 李德仁, 王树良, 史文中, 王新洲. 论空间数据挖掘和知识发现. 武汉大学学报, 2001, 26(6): 491~497
- 2 邱凯昌, 等. 空间数据挖掘与知识发现. 武汉大学出版社, 2001
- 3 Ester M, Kriegel H P, Sander J. Spatial Data Mining. A Database Approach. In: Proc. 5th Int. Symp. on Large Spatial Database, Berlin, Germany, 1997
- 4 Ester M, Frommelt A, Kriegel H P, et al. Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support. Data Mining and Knowledge Discovery, 2000
- 5 Koperski K, Han J, Adhikary J. Mining Knowledge in Geographical Data. In: communications of ACM. 1998
- 6 Ester M, Frommelt A, Kriegel H P, et al. Algorithms for Characterization and Trend Detection in Spatial Database. In: Proc. 4th Intl. Conf. on Knowledge Discovery and Data Mining, New York, 1998